Reasoning over Knowledge-based Generation of Situations in Context Spaces to Reduce Food Waste

Niklas Kolbe[1], Arkady Zaslavsky[1,3], Sylvain Kubler[2], and Jérémy Robert[2]

[1] Commonwealth Scientific and Industrial Research Organisation, Data61, Clayton VIC 3168, Australia,
{niklas.kolbe, arkady.zaslavsky}@csiro.au

[2] University of Luxembourg, Interdisciplinary Center for Security, Reliability and Trust, L-2721 Luxembourg, Luxembourg,
{sylvain.kubler, jeremy.robert}@uni.lu

[3] Saint Petersburg National Research University of ITMO, 49 Kronverksky Pr., St. Petersburg, 197101, Russia,
arkady.zaslavsky@acm.org

Abstract. Situation awareness is a key feature of pervasive computing and requires external knowledge to interpret data. Ontology-based reasoning approaches allow for the reuse of predefined knowledge, but do not provide the best reasoning capabilities. To overcome this problem, a hybrid model for situation awareness is developed and presented in this paper, which integrates the Situation Theory Ontology into Context Space Theory for inference. Furthermore, in an effort to rely as much as possible on open IoT messaging standards, a domain-independent framework using the O-MI/O-DF standards for sensor data acquisition is developed. This framework is applied to a smart neighborhood use case to reduce food waste at the consumption stage.

Keywords: Situation Awareness, Context Awareness, Pervasive Computing, Ontologies, Internet of Things, Context Space Theory

1 Introduction

The Food and Agriculture Organization of the United Nations (FAO) estimates that up to 50% of produced food is wasted all over the world, which has a non-negligible impact on the society, environment and economy (e.g., starvation, carbon emission and economic cost, *etc.*) [8] [9]. The information and system intelligence and analytics capabilities enabled by pervasive environments and the so-called Internet of Things (IoT), could potentially help drive innovative sustainable development and business models. The IoT offers provisions for real-time analysis on any operation or process as a game-changer when it came to creating environmental benefits. To take full advantage of the IoT, it is nonetheless crucial not to focus only on sensor data, but also on the "context" in which this data was generated, monitored, and so forth. Context is any information that characterizes the environment of an entity (a person, group of person, a place or a Thing) relevant to the interaction of the application and end-users [1]. Context-awareness means understanding the whole environment and current situation of the entity. Context can be processed to developed more advanced services such as "situation-awareness", which can be seen as a course of events that evolves to more sophisticated relations between entities (or even situations).

Situations are defined as external semantic interpretation of sensor data on a higher level of abstraction than activities or context [20]. Thus, situation awareness strongly depends on expert knowledge to interpret sensed data. A situation aware approach requires modeling, reasoning, and sensor data acquisition, while considering several functional requirements for each step. Defining expert knowledge, adopting reasoning engines and integrating sensor data are extensive and error-prone tasks, complicating the development of situation aware applications. To address this problem and easily capture all domain- and application-specific dependencies, this paper investigates a general ontology-based framework for situation awareness based on standardized technologies.

The paper is structured as follows: Sections 2 and 3 present the background and related work in the area. Section 4 proposes the core ontology and the framework architecture for general situation awareness. Section 5 presents a use case of a smart neighborhood to reduce food waste and an evaluation of the proposed framework; the conclusion follows.

2 Background

Several theories (e.g., Context Space and Situation Theory, Semantic Sensor Network) and technological building blocks (e.g., O-MI/O-DF standards) have been considered to design the proposed framework for general situation awareness. This section therefore provides the necessary background regarding each of these theories and technologies.

First, the use of ontological approach allows situation modeling with rich semantics that can be understood and shared among humans and machines. In the life sciences community, Ontology Web Language (OWL) is extensively used and has become a de facto standard for ontology development [13]. OWL provides a vocabulary for the Resource Description Format (RDF) by extending the RDF Schema vocabulary. Given this, our framework is designed based on the OWL standard.

Another key theory considered in our framework is the Situation Theory Ontology (STO) [10], which was developed based on situation semantics referred to as Situation Theory [7]. This theory will be applied to model situations, where facts of situations are formulated as "infons", and "situations" are defined by specifying which infons they support (see Eq. 1). An infon is a relation of n objects, whereas objects can be individuals, attributes or situations. The polarity (0/1) specifies whether this relation is true or false.

$$S \models \ll \mathit{relation}, a_1, ..., a_n, 0/1 \gg \tag{1}$$

The Semantic Sensor Network (SSN) ontology [6] is a standard ontology to represent knowledge about a sensor network (e.g., sensing devices, measured properties and deployed platforms. . .), without initially taking into consideration actuators. The Semantic Actuator Network (SAN) ontology [15] was further developed as a counterpart to SSN. Both ontologies will be considered in our framework to specify the system setup.

The Context Space Theory (CST) [14] was developed – *based on a spatial representation of context* – to provide a general context model with a rich theoretical foundation. The context space is defined through context attributes and situations are modeled as subspaces. By combining specification- and learning-based techniques, and by supporting algebraic operations, CST allows for general reasoning about situations. CST-based reasoning is implemented in ECSTRA [4] with a flexible architecture for situation aware systems, which will be applied for the implementation of this study. The knowledge defined in STO will be used to generate the situation spaces in the context space.

Finally, to increase interoperability of the framework in a range of IoT settings, recent IoT messaging standards published by The Open Group, namely the O-MI (Open-Messaging Interface) and O-DF (Open-Data Format) standards, are used to enable peer-to-peer data exchange between different systems and devices [18]. O-MI messages can be exchanged on top of well-known protocols like HTTP, SOAP or SMTP, while O-DF [17] is a generic content description model for Things in the IoT, which can be extended with more specific vocabularies (e.g., using domain-specific ontology vocabularies). The knowledge defined in SSN and SAN will be used to generate context collectors based on O-MI/O-DF.

3 Related Work

Besides STO, other upper ontologies for situation awareness were developed by the research community. In the Core SAW Ontology [12], situations are represented as a set of entities with attributes, goals and foremost relations. It furthermore integrates observed sensed data in the ontology. The Situational Context Ontology [2] starts from a context perspective and adds a situational structure around it, while offering provisions for modeling imprecise sensor data (using fuzzy logic). The Situation Ontology developed in [19] is based on a context and situation layer, and allows the definition of atomic and composite situations based on context values.

Several hybrid approaches, combining ontologies with other reasoning techniques to achieve situation awareness have been proposed. In [5] the feasibility of integrating ontological knowledge into CST has been shown, based on both a context ontology and rule-based situation definitions. Situation spaces are generated by processing the rules and querying the ontology with SPARQL. The Wavellite framework [16] was proposed to achieve situation awareness in environmental monitoring. It uses upper ontologies, including STO, as a knowledge base and combines it with rules and neural networks for inference. However, the reasoning engines are application-specific. A number of approaches add rule-based reasoning around an ontology, such as BeAware! [3], which proposes a general reasoning technique by extending the Core SAW Ontologies and including relation types.

The aforementioned approaches have not been previously used in a food waste reduction or management process. We could nonetheless point out a few community-based social networks that apply pervasive computing to address this challenge, such as EUPHORIA (standing for Efficient food Use and food waste Prevention in Households through Increased Awareness)[11], which is a project that allows users to log and track their everyday food related behavior and redirect these, through social influence, towards more sustainable food related practices. Nonetheless, the project has focused on social behavior around food consumption (necessitating manual inputs via a mobile application), and has not proposed any IoT-based services to automate the discovery of food in the neighborhood that is e.g. close to its expiry date, and propose to end-users appropriate recipes. The paper investigates and develops a framework that fulfills such IoT-based services.

4 Framework Design

This section presents the core ontology for CST, generation of situation spaces and the framework architecture. In this respect, sections 4.1 and 4.2 respectively detail the CST Ontology, and how situations are generated based on this ontology. Section 4.3 provides an "at a glance" overview of the the overall proposed framework.

4.1 CST Ontology

Situation spaces in CST are defined through a set of acceptable regions for all context attributes. Each context attribute is assigned with a relevance weight $w_i \in [0, 1]$. Furthermore acceptable regions are assigned with a contribution function η_i^S, which assigns a contribution $\in [0, 1]$ to each value within the acceptable region. The overall confidence if a situation is occurring is calculated based on the relevance and the contribution for a context state x, as shown in Eq 2.

$$\mu_s = \sum_{i=1}^{n} w_i * \eta_i^S (x_i) \tag{2}$$

For final inference, the confidence value is compared to a threshold ε_i, as formulated in Eq 3.

$$\gamma = (\mu_s \geq \varepsilon_i) \tag{3}$$

The specifications are the requirements for situation modeling. Thus, concepts of STO and SSN/SAN were mapped and extended with CST specific information. The core of the CST ontology is shown in Fig. 1. STO and SSN/SAN were mapped through two key connections: (i) `sto:Attribute` is defined as

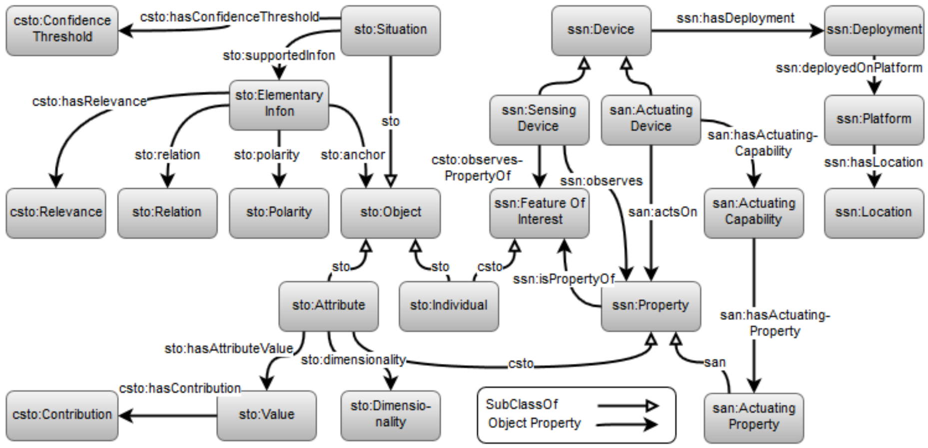

Fig. 1. Core Ontology for Context Space Theory

subclass of `ssn:Property`, and (ii) `sto:Individual` is defined as subclass of `ssn:FeatureOfInterest`. Through these definitions, sensors are observing attributes of individuals in STO situation definitions. Context attributes in CST correspond to both `sto:Attributes` and `ssn:Properties`. An acceptable region can be defined as a `sto:Value` of an attribute, whereas the value is specified as an interval. The ontology was extended to capture further CST-related concepts, which includes the `csto:ConfidenceThreshold` for a situation, `csto:Relevance` for an infon[4] and `csto:Contribution` for values of acceptable regions.

4.2 Situation Generation

Situation generation from the core ontology can be declined into three categories, namely based on (i) situation objects, (ii) situation types, and (iii) situation objects generated by type definitions. Individual situations in STO are modeled with concrete instances for situations, infons and other objects. The objects involved in a situation may be application dependent, which makes it unable to reuse the situation specification. Instead, situation types can be defined based on OWL class axioms. Listing 1.1 presents an example for a situation type definition to infer if a person is running. The situation supports two infons, which are based on the movement speed and the heart rate of a person. The complete definition (not shown in the listing for simplicity) further includes the specification of the attributes, acceptable regions and contribution.

Listing 1.1. Example for Situation Type Definition in the Ontology

```
PersonRunning owl:equivalentClass (
  sto:Situation
```

[4] The relevance belongs to context attributes, but an attribute can have a different weight for different facts about situations.

```
  and (sto:supportedInfon value HighHeartRateInfon)
  and (sto:supportedInfon value FastMovementInfon)
  and (sto:relevantIndividual value Person)
  and (csto:hasConfidenceThreshold value 0.8)
)
HighHeartRateInfon owl:equivalentClass (
  sto:ElementaryInfon
  and (sto:relation value Heartrate)
  and (sto:anchor1 value Person)
  and (sto:anchor2 value HighHeartRateAttribute)
  and (sto:polarity value _1)
  and (csto:hasRelevance value 0.6)
)
FastMovementInfon owl:equivalentClass (
  sto:ElementaryInfon
  and (sto:relation value Movement)
  and (sto:anchor1 value Person)
  and (sto:anchor2 value FastMovementAttribute)
  and (sto:polarity value _1)
  and (csto:hasRelevance value 0.4)
)
```

The algorithms to generate situation spaces iterate over the given situation and infon definitions (for both objects and types), retrieve the corresponding information about the context attributes, acceptable regions, contribution, *etc.*, and resolve dependencies to subspaces. Algorithm 1 shows the generation of a situation space for one situation definition. Each generated situation space is then added with its confidence threshold to the context space.

Situation types ease the modeling process because situation definitions do not depend on application specific objects. In CST, objects can share the same situation space, while each object maintains a different state in the context space. The origin of the state may come from different sensors for different objects. This is captured and maintained through the integration of the SSN ontology. If it is not desired to resolve these dependencies via the ontology, separate situation spaces can be generated for each relevant individual involved in the situation (case (iii)).

4.3 Framework Architecture

Overall, the architecture needs to integrate the following major building blocks for a situation aware system:

- Knowledge base (CST ontology)
- Ontology management (OWL API, SPARQL-DL API, Pellet, Protégé)
- CST-based reasoning (ECSTRA)
- Sensor data acquisition (IoT Data Server for O-MI agents)
- Client application

Algorithm 1 Generation of Situation Space

```
function GenerateSituationSpace(situation)
    situationSpace ← new SituationSpace(situation.name);
    for all situation.getInfons() do
        for all infons.getAnchors() do
            if anchor.type() == attribute then
                axis ← newAxis(attribute.name)
                for all attribute.getAcceptableRegions() do
                    if infon.polarity() == 1 then
                        axis ← addRegion(value, contribution)
                    else
                        axis ← addAsymmetricRegion(value, contribution)
                situationSpace ← addAxis(axis, infon.getRelevance())
            else if anchor.type() == situation then
                SubSpace ← GenerateSituationSpace(anchor)
                situationSpace ← addAxisSubSpace(SubSpace)
            else if anchor.type() == individal then
                                        ▷ Not considered in CST Situation Spaces
    return situationSpace
```

Fig. 2 illustrates the complete architecture designed in our study to integrate those building blocks. The knowledge base consists of CSTO-based application ontologies. Multiple ontologies with different situation specifications and the application setup can be provided for the system (ontology editors like Protégé can be used in this respect). The ontology management component is responsible for the programmatic access and manipulation of the knowledge base. Since the algorithm needs to access the TBox axioms (terminology) of the ontology, an OWL-centric approach is preferred over a RDF-centric approach. Tools used in our framework include OWL API, SPARQL-DL API and the Pellet reasoner. The ECSTRA implementation is used for CST-based reasoning. Finally, the O-MI/O-DF standards are integrated, meaning that instead of subscribing to a central publish/subscribe engine, the context collectors subscribe directly to one or more O-MI nodes and receive the notifications in an O-DF payload format. The central manager (*cf.* Fig. 2) forms an interface to integrate and coordinate all these components, while providing a facade to client applications to initialize the system and send enhanced reasoning requests. At a more concrete level, the tasks of the manager are:

1. Loading and merging given ontologies.
2. Initializing the ontology reasoner.
3. Generating situation spaces based on object and type definitions.
4. Generating O-MI context collectors based on the given specifications.
5. Initialize the application space.
6. Resolving dependencies to individuals and sensors from reasoning requests.
7. Distributing reasoning results.

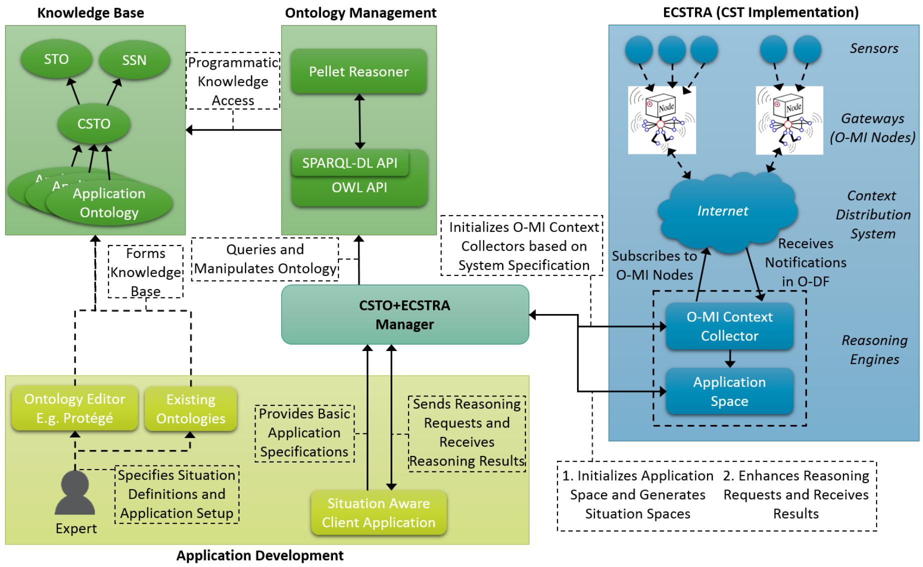

Fig. 2. Framework Architecture for Situation Awareness

From an operational perspective, the overall architecture has been implemented as a JVM-based library, which can be deployed in an agent-based architecture. Context collectors can potentially be added manually to access other types of information sources.

5 Use Case: Reducing Food Waste

This section describes a proof-of-concept and an evaluation of the proposed framework. Firstly, the situation awareness framework is applied to a use case to reduce food waste in a smart neighborhood in section 5.1. Subsequently a discussion of the features of the framework and a performance evaluation follows in section 5.2.

5.1 Use Case Scenario and Implementation

The overall scenario is depicted in Fig. 3, which considers a connected neighborhood and exploits situation awareness to give best recommendations about the consumption of food items (e.g. relevant recipes, incentives for food sharing. . .). The application is developed as a JVM-based web application, where recipes are requested from an open REST API[5]. It should be noted that our implementation is based on the following assumptions:

[5] Yummly Recipe API: https://developer.yummly.com/

– The implementation is based on a simulated smart neighborhood, composed of three households.
– Each household generates (simulated) sensor data values. To sense information about food items, it is assumed that each item is labeled with an RFID tag and smart fridges are equipped with RFID readers, to read these tags when items are placed inside the fridge.
– Information stored in the RFID tags includes the available amount of items and related expiration date.
– Sensor data providing information about when and how to access food items is simulated. For example, this input can be simulated based on human being's activity in the household or on the availability of smart access devices (e.g., smart locks like slock.it[6]).
– All sensor values are published through an IoT/neighborhood avatar (an O-MI node in our case) that aggregates and publishes neighborhood-related information in a standardized manner.

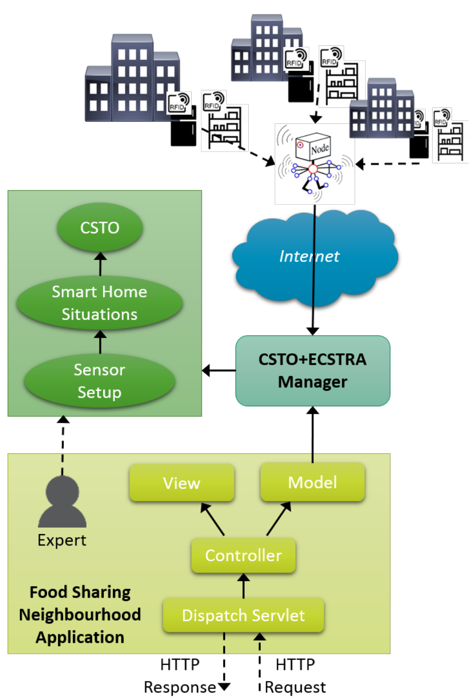

Fig. 3. Use Case Architecture

[6] Smart Locks slock.it: https://slock.it

The recommendation for consumption of a food item is based on the shelf life and relative amount of available stock. Eq. 4 shows the situation type definition in situation theory that was modeled in the CST ontology, where parameters $\dot{f}$, $\dot{e}$ and $\dot{s}$ respectively stand for food items, close expiration dates and relative high stock. Acceptable regions can be modeled with fuzzy sets, e.g. context collectors can fuse sensed data to *low*, *medium* and *high* amount of available stock.

$$\left[\dot{S_R} | \dot{S_R} \models \ll expires, \dot{f}, \dot{e}, 1 \gg \wedge \ll stock, \dot{f}, \dot{s}, 1 \gg\right] \tag{4}$$

Listing 1.2 shows an OWL individual definition of the ontology for a sensor. It specifies an RFID reader, which is capable to observe different attributes (*Expiration* and *Amount*) of instances of the class *Fooditem*, which in turn is part of the situation type definition presented in Eq. 4. It is attached to a specific household via the `ssn:hasLocation` object property.

Listing 1.2. Example of Sensor Modeling

```
RFIDSensor001 rdf:type FridgeRFIDSensor
RFIDSensor001 ssn:observes Expiration
RFIDSensor001 ssn:observes Amount
RFIDSensor001 ssn:hasLocation Household1
RFIDSensor001 csto:observesPropertyOf Fooditem
```

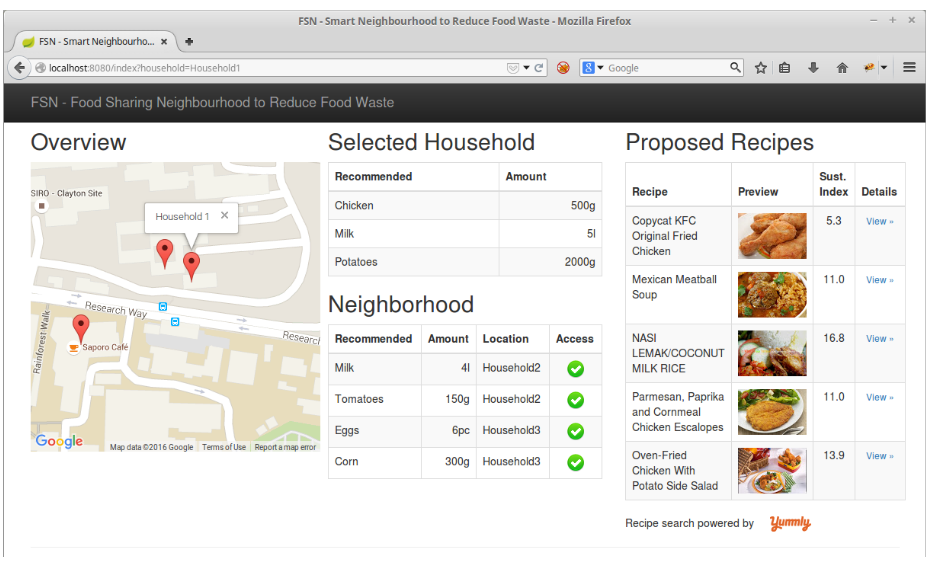

Fig. 4. Food Sharing Neighborhood Application

Fig. 4 shows a screenshot of the web application used in our neighborhood waste management system, along with the recommendation outputs. The system identifies the items recommended for usage and displays the available amount, location and current accessibility. Furthermore it shows recipes that can be cooked with the available food items. A sustainability index is calculated for each recipe, based on the consideration of ingredients that are recommended for usage and the amount that will be prevented from being wasted. The index takes into account the environmental impact of the commodity group (carbon footprint, blue water footprint, economic cost [8]) of each ingredient.

5.2 Evaluation and Ontology Performance

The framework presented in this paper is based on a rich foundation for both situation modeling and situation reasoning, whose key functionalities are:

- **Situation modeling.** Space and time aspects, situation types, roles of objects, relations.
- **Knowledge.** Integrating knowledge about situations and systems, allowing reuse and sharing with semantic web technologies.
- **Reasoning.** General applicability, uncertainty and temporal aspects, can be extended with prediction and proactive adaption. Enhanced through consideration of involved individuals via STO.
- **Application development.** Automated integration of sensor data acquisition, less complex for deployment.

After the automated initialization of the system, situation reasoning can be performed directly with ECSTRA or with enhanced requests involving access to the ontology. The proposed architecture does not the capabilities of the existing ECSTRA implementation. To perform (optional) enhanced reasoning requests, access to the ontology during run-time is required. Fig. 5 shows the added computation time to resolve dependencies to individuals, attributes, situations and sensors for reasoning requests via the ontology.

The ontology was populated with test data. The largest data set consisted of 7500 situation definitions with attached infons, attributes etc. which corresponded to 137400 axioms in the ontology. The figure shows the average computing time with standard derivation for 1000 test runs per data set.

The test indicates a complexity of $O(n)$. However, ontology reasoners demand high memory. With further increased testing data the available heap space (6GB) was not sufficient. This might be an issue for very large-scale systems. In this case run-time reasoning requests should be sent directly to individual specific generated situation spaces.

6 Conclusion and Future Work

The selected approach was based on an intensive study of situation aware approaches. As a result of this discussion, the combination of Situation Theory and Context Space Theory was motivated by automatic generation of Situation

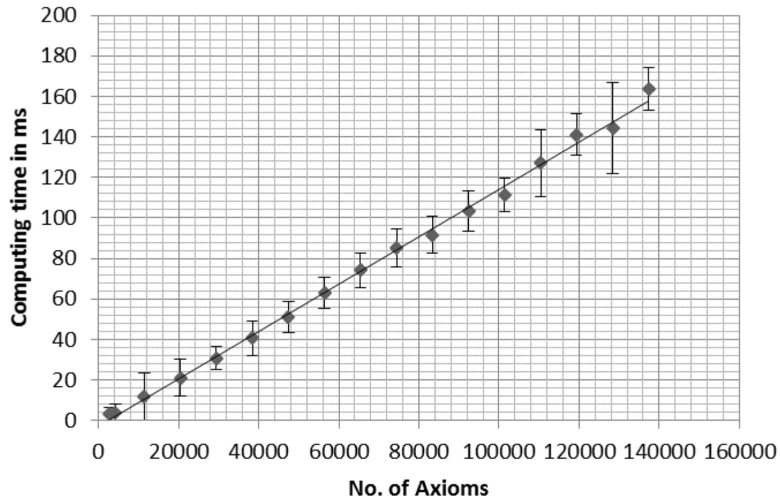

Fig. 5. Performance Results for Ontology Access during Reasoning Requests

Spaces in CST with knowledge specified in an ontology. By identifying requirements for a holistic framework, STO, SSN and SAN were combined and extended to serve as a core ontology for a CST-based system. Algorithms to extract the knowledge from the ontology and initialize the application space were proposed.

Further discussion led to a design of an overall framework based on the proposed core ontology and ECSTRA for CST-based reasoning. In order to meet the requirements for platform independent sensor data acquisition, the IoT standards O-MI/O-DF were integrated into the system. The contribution of this work is a Java library which was designed to allow an efficient use of these components to develop situation aware applications.

As a proof-of-concept the framework was applied to a use case, which validated the feasibility of the approach. By showcasing a system to reduce food waste at consumption stage the use case demonstrated the enabling effects of situation aware systems regarding the contribution to sustainability.

Further work identified includes the consideration of a dynamic environment (joining and leaving objects, discovery of new sensor sources), validation of situation occurrences based on OWL axioms specified in the ontology, generating situation spaces with incomplete knowledge and adding actuation to the situation aware framework.

Acknowledgments. Authors acknowledge support from EMM PERCCOM, IoT EPI bIoTope Project, which is co-funded by the European Commission under H2020-ICT-2015 program, Grant Agreement 688203, as well as financial support from Ministry of Science & Education of Russian Federation, Grant 14.587.21.0031.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Handheld and ubiquitous computing. pp. 304–307. Springer (1999)
2. Anagnostopoulos, C.B., Ntarladimas, Y., Hadjiefthymiades, S.: Situational computing: An innovative architecture with imprecise reasoning. Journal of Systems and Software 80(12), 1993–2014 (12 2007)
3. Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., Schwinger, W.: Beaware!situation awareness, the ontology-driven way. Data Knowledge Engineering 69(11), 1181–1193 (2010)
4. Boytsov, A.: Ecstra distributed context reasoning framework for pervasive computing systems. Smart Spaces and Next Generation Wired/Wireless Networking (2011)
5. Boytsov, A., Zaslavsky, A., Eryilmaz, E., Albayrak, S.: Situation Awareness Meets Ontologies: A Context Spaces Case Study, pp. 3–17. Modeling and Using Context, Springer (2015)
6. Compton, M., Barnaghi, P., Bermudez, L., GarcA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A.: The ssn ontology of the w3c semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web 17, 25–32 (2012)
7. Devlin, K.: Situation theory and situation semantics. Handbook of the History of Logic 7, 601–664 (2006)
8. FAO: Food wastage footprint: Impacts on natural resources. Tech. rep., FAO (2013), http://www.fao.org/docrep/018/i3347e/i3347e.pdf
9. Gustavsson, J., Cederberg, C., Sonesson, U., Otterdijk, R.V., Meybeck, A.: Global food losses and food waste. Food and Agriculture Organization of the United Nations (2011)
10. Kokar, M.M., Matheus, C.J., Baclawski, K.: Ontology-based situation awareness. Information fusion 10(1), 83–98 (2009)
11. Lim, V., Yalva, F., Funk, M., Hu, J., Rauterberg, M.: Can we reduce waste and waist together through euphoria? In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on. pp. 382–387. IEEE (2014)
12. Matheus, C.J., Kokar, M.M., Baclawski, K., Letkowski, J.A., Call, C., Hinman, M.L., Salerno, J.J., Boulware, D.M.: Sawa: An assistant for higher-level fusion and situation awareness. In: Defense and Security. pp. 75–85. International Society for Optics and Photonics (2005)
13. McGuinness, D.L., Harmelen, F.V.: Owl web ontology language overview. W3C recommendation 10(10), 2004 (2004)
14. Padovitz, A., Loke, S.W., Zaslavsky, A.: Towards a theory of context spaces. In: Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on. pp. 38–42. IEEE (2004)
15. Seydoux, N., Alaya, M.B., Drira, K., Hernandez, N., Monteil, T.: San (semantic actuator network), https://www.irit.fr/recherches/MELODI/ontologies/SAN.html

16. Stocker, M., Ronkko, M., Kolehmainen, M.: Situational knowledge representation for traffic observed by a pavement vibration sensor network. Intelligent Transportation Systems, IEEE Transactions on 15(4), 1441–1450 (2014)
17. The Open Group: Open data format standard (o-df) (Open Group Standard) (2014), https://www2.opengroup.org/ogsys/catalog/C14A;
18. The Open Group: Open messaging interface technical standard (o-mi) (Open Group Standard) (2014), https://www2.opengroup.org/ogsys/catalog/C14B
19. Yau, S.S., Liu, J.: Hierarchical situation modeling and reasoning for pervasive computing. In: Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on. p. 6 pp. IEEE (2006)
20. Ye, J., Dobson, S., McKeever, S.: Situation identification techniques in pervasive computing: A review. Pervasive and Mobile Computing 8(1), 36 – 66 (2012)